Management's Inspection Responsibilities and Tools for Success

Roger Stewart and Lew Priven
Stewart-Priven Group

There are many pitfalls that cause software inspections to fail. This article addresses management's critical role in preventing these pitfalls to attain successful inspections. In addition to meeting their responsibilities, management needs a comprehensive computerized set of tools to support their efforts. By carrying out their responsibilities—supported by inspection-specific tools—management will be better equipped to implement sustained successful project inspections that consistently reap the benefits of lower project cost and high product quality.

For this article, an inspection is defined as a preemptive systematic peer review of work products by three to five trained individuals (e.g., stakeholders) using a well-defined and documented process. The goal of peer review is to reduce project rework cost and raise product quality and return on investment (ROI) by detecting and removing (non-trivial) defects as early as possible in the software development life cycle (SDLC) or closest to the points of defect insertion.

The objective of this article is to explore upper (middle and senior) management's software inspection responsibilities and how computerized inspection tool features can improve meeting those responsibilities for consistent inspection success.

Ten Common Inspection Pitfalls

Figure 1 identifies ten common inspection pitfalls that cause software inspections to fail. It is the responsibility of upper management to solve or prevent each pitfall. These pitfalls were the focus of our January 2008 [1] CrossTalk article. This article is a follow-up and in the future we expect to explore inspection tool features in more depth than is discussed here.

Having an adequate SDLC infrastructure is a prerequisite for inspection success. Specifically, company culture needs an enabling SDLC infrastructure where the disciplines of planning, scheduling, data collection, monitoring, and tracking are already ingrained in the culture. The CMM® and its successor CMMI® identify these disciplines as Level 2 management skills. Inspections are dependent upon these underlying disciplines and thus require Level 2 management skills as the foundation for success. We have observed that companies typically encounter multiple inspection pitfalls, any one of which can result in not achieving the lasting benefits that inspections can provide when properly implemented.

This article addresses management's responsibilities in eliminating the pitfalls and explains how inspection tools can help make this easier to achieve.

Management's Understanding of Inspections

Prevention of the inspection pitfalls shown in Figure 1 requires management to first understand—and then fulfill—their inspection responsibilities. Typically when project inspections are introduced, the focus on management's critical role is either overlooked or not given the attention required. Just as inspection practitioners (inspectors) receive training on inspection process execution, management needs complementary training that focuses on identifying and achieving their inspection responsibilities.

The cost and schedule impact of inspections are primarily borne by the requirements, design, and coding areas, although these areas realize some of the reduced cost and higher-quality inspection benefits. The majority of inspection benefits are realized in testing and maintenance, requiring management to have a life-cycle view of the project. Therefore, management at all levels and across all development phases (including upper management in areas that will not use inspections), require inspection education.

Management instruction on how to manage inspections occurs prior to practitioner training and focuses on the responsibilities of management for achieving inspection success, specifically:

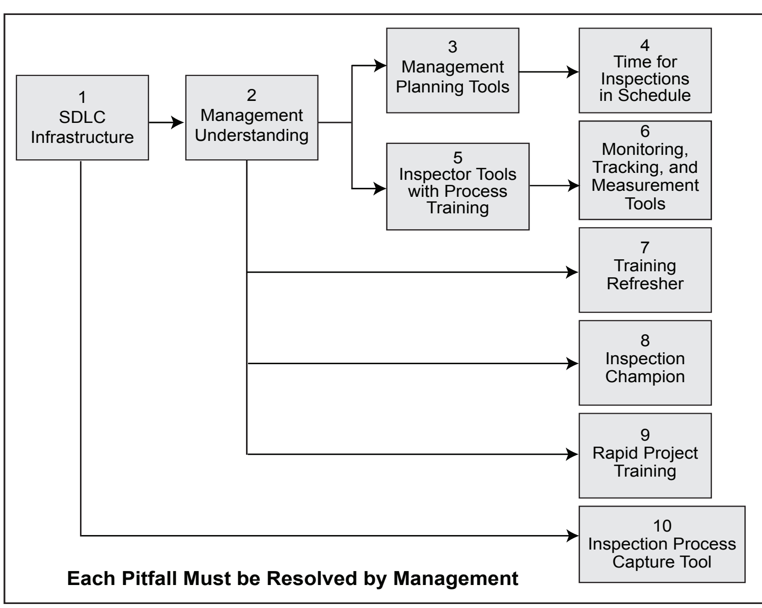
- Prevention of the 10 identified inspection pitfalls.
- Effective use of management support tools.

Management's Inspection Responsibilities

There are seven stages for successfully managing inspections, which are identified in Figure 2. These stages are:

- **Stage 1: Project Planning and Stage 2: Savings Estimation.** Management identifies the number of inspections required by the project, their cost, and the estimated net project savings from implementing the inspections. These two stages occur prior to conducting any project inspections.
- **Stage 3: Commitment.** Also occurring prior to conducting any project inspections, management establishes and incorporates inspection planning and control

Figure 1: *Managing Software Inspection Pitfalls*



® CMM and CMMI are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

procedures into the project's plans. Elements of the commitment stage are described later in the article.

- **Stage 4: Execution.** Management's principal responsibility in this stage is to provide inspection tools that facilitate correct and consistent (repeatable) execution of inspections by teams, organizations, and locations supporting their project. Additionally, these inspection tools should automate the collection of inspection data for later monitoring, tracking, and measurement of inspections results.
- **Stage 5: Monitoring.** Management assesses the interim inspection process conformance and results. If either appears questionable, then consultation with the inspection team leader may reveal that a partial or full reinspection is needed before further time is invested investigating and fixing the potential problems that have been uncovered. This stage also provides early identification of potential defect-prone areas and the need for improvement to inspection materials, team selection, and inspection process adherence.
- **Stage 6: Tracking.** After each inspection exit, individual inspection results are collected and consolidated into multiple inspection totals used for tracking to-date project savings against the stage 2 savings estimate. Tracking also includes the ongoing trend analysis of defect reason and origin metrics collected during inspections, which are used for future defect prevention improvements to development processes.
- **Stage 7: Measurement.** This stage occurs throughout the development life cycle, providing for evaluation of early defect removal by inspections, defect removal by subsequent testing, and any other means of defect removal (e.g., pretest automated code analysis tools). This stage also provides for the evaluation of the effectiveness of the inspection implementation for pre-test defect removal at or close to the points of insertion. It is also a means to assess progress toward the defect removal goals recorded in the project's quality plan (described later).

These stages should not be confused with the seven inspection steps typically associated with performing inspections [2], which occur during the execution and monitoring stages of management's inspection responsibilities (see Figure 2).

With this understanding of management's responsibilities, let's take a detailed look at how a set of inspection management tools might be used.

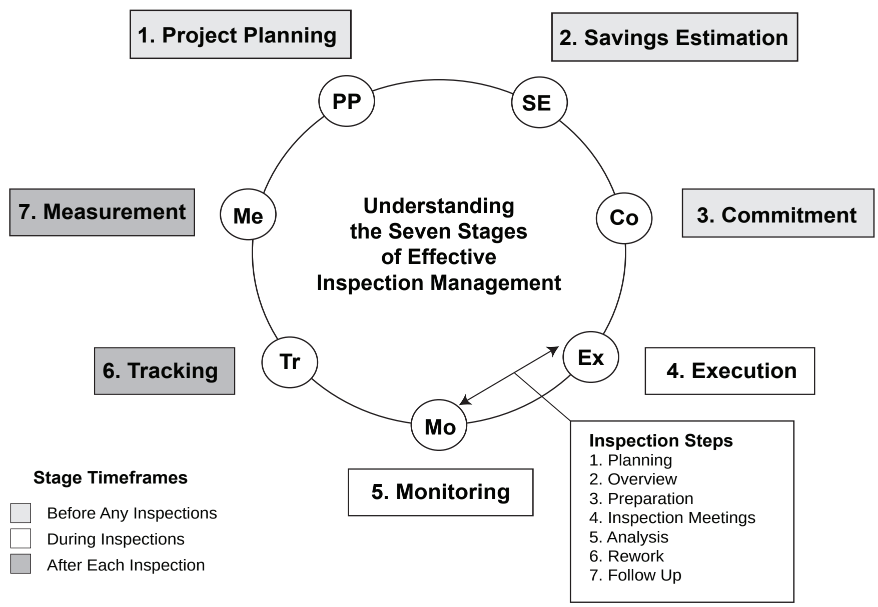


Figure 2: *Stages of Management's Inspection Responsibility*

Management Inspection Tools

As previously stated, without management tools designed for each inspection stage and integrated with each other to build upon inspection information collected in earlier stages, management will be challenged to meet their inspection responsibilities and achieve project success.

Figure 3 identifies what a set of computerized inspection tools might consist of. The tools shown are grouped according to the four timeframes where they would be used:

- Before Any Inspections (during project planning).
- During Inspections (execution and monitoring).
- After Each Inspection (tracking).
- Throughout Development and Testing (measurement).

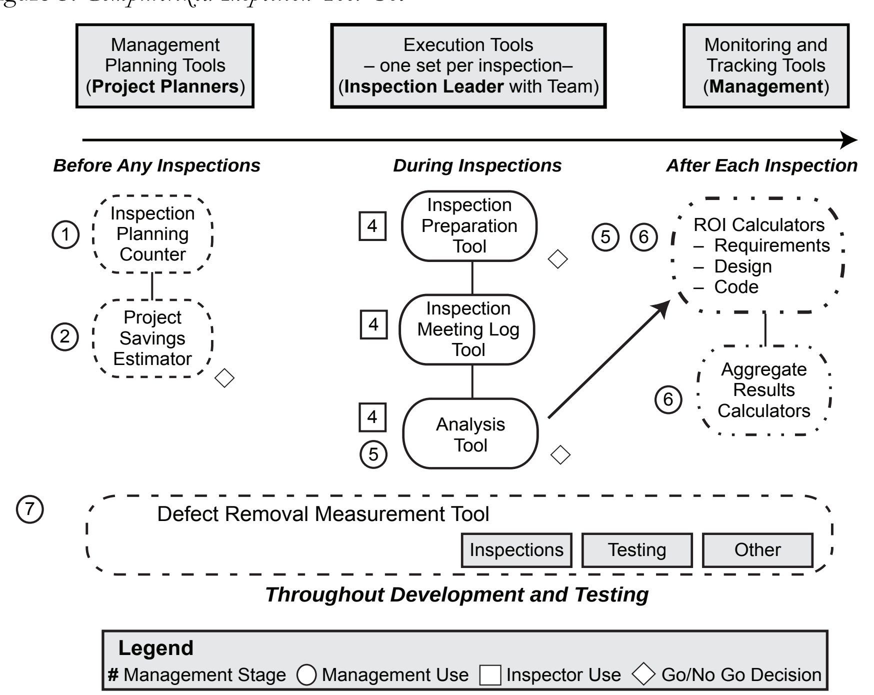
The legend in Figure 3 shows the association of each tool with the inspection stage they apply to (Figure 2), whether the tool is for use by management or by inspection practitioners, and where the three tool-aided decision points are for proceeding with an inspection.

Inspection Tool Use

Before Any Inspections

During the first stage (project planning), a planning counter tool should be used to compute the number of inspections to be conducted during a project, the estimated

Figure 3: *Computerized Inspection Tool Use*



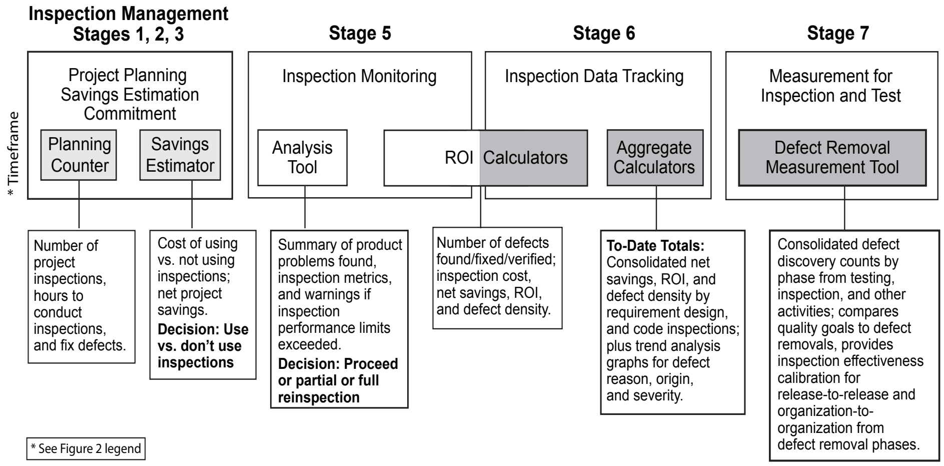


Figure 4: *Overview of Management's Inspection Tool Features*

number of defects to be removed, and the projected cost for conducting project inspections to find and fix those defects.

During the savings estimation stage, a savings estimator tool would then use the planning counter tool output along with historical data and industry data to estimate the net savings (or cost) the project would realize (from the inspections and the resulting reduced rework and testing). Using this data, management can now make an informed decision whether to employ inspections.

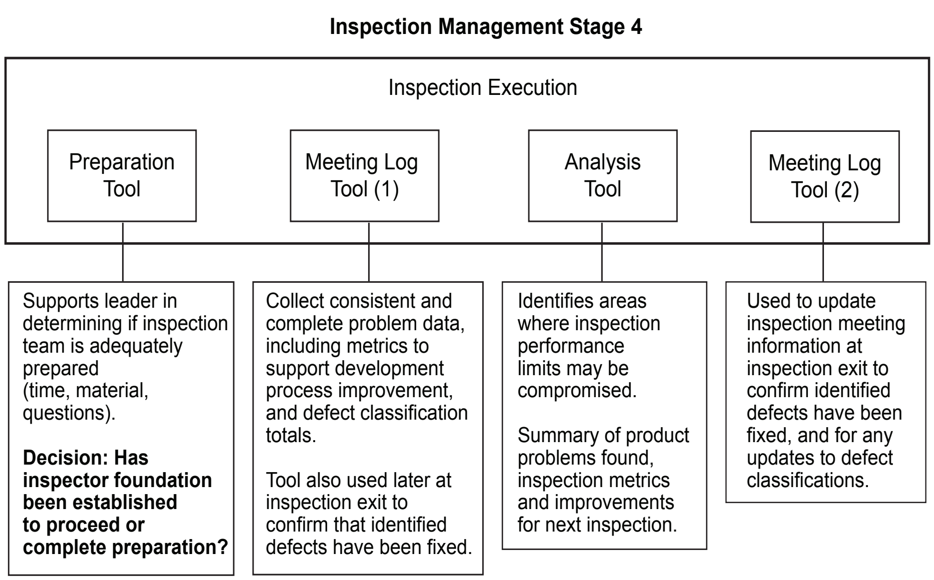
Our experience is that few project managers will really commit to inspections without knowing the net savings benefit from implementing inspections. Few organizations (if any) know how to compute their true inspection cost and, more importantly, accurately estimate their net project savings from implementing inspections as a primary means of raising product quality and lowering both development and maintenance costs. Using both planning counter and savings estimator tools can provide management with the ability to compute the net project savings estimate.

Computerized inspection tools are a critical aid to management for both of the first two stages. Furthermore, if these tools are parameterized to accept historical enterprise data and possible future development and testing profiles, then management has the additional flexibility to examine what-if scenarios that lead to making an informed decision for their project to use, or not use, inspections. This information can also be used to sell or defend their decision, as needed.

The commitment stage—performed prior to conducting any project inspections—includes the following responsibilities:

- Conveying a strong net-savings mindset (as opposed to a cost focus) so often missing from projects trying to do inspections.
- Securing buy-in from upper management across all project areas (e.g., requirements, design, code, and test).

Figure 5: *Summary of Practitioner Inspection Tools*



The estimated net savings output of a savings estimator tool is ideal for accomplishing these responsibilities.

The other elements of inspection commitment responsibilities can be accomplished without additional inspection tools by:

- Establishing a quality plan with quantifiable defect removal goals for each development phase (requirement definition through testing and customer use).
- Developing project plans that contain individual inspection schedules.
- Developing inspection budgets and/or setting up cost accounts for tracking inspection cost.
- Scheduling upper management inspection awareness classes.
- Securing rapid practitioner training that doesn't have delays between multiple practitioner classes (a characteristic of many company training departments).
- Employing an inspection methodology (see [1]) that resolves all inspection pitfalls in addition to accomplishing both management and practitioner inspection training.

Another aspect of management commitment is to provide an inspection environment that facilitates inspection practitioners in attaining the full quality and cost savings benefits of inspections. To accomplish this, an inspection tool set should be obtained for practitioners that:

- Assists inspector adherence to both the inspection process [2] and performance limits that characterize successful inspections.
- Provides warnings where proceeding to the next inspection step could be risky.
- Collects complete and consistent inspection data for inspection monitoring and savings tracking.
- Provides a snapshot report for monitoring inspection process adherence and interim pre-fix inspection results.
- Provides defect metrics for the post-inspection pursuit of development process improvements for future defect prevention.

During Inspections

Managers of inspection practitioners do not attend inspection meetings, but management needs the capability to monitor interim results at the conclusion of inspection meetings (step 4 in Figure 2). This capability must be uniform and repeatable regardless of what team, organization, or location is conducting project inspections.

Ideally, a one-page snapshot from the tool output of the inspection analysis step will provide management with the data needed to meet their inspection monitoring responsibility. This would include identifying where inspection performance thresholds have been exceeded. This will assist management in determining whether a partial or full reinspection is needed. Inspection performance thresholds might include the number of lines inspected, inspection meeting length, preparation rate, inspection rate, number of inspectors, number of inspectors with adequate domain knowledge and language knowledge, adequacy of inspection materials to meet inspection entry criteria, and adequacy in individual inspector preparation. While some threshold violations may be suspected or possibly known prior to an inspection meeting (e.g., preparation rate), they should also be included on a post-inspection meeting report for management to get the complete picture on whether to allow the inspection to move forward into rework (step 6 in Figure 2) or whether some form of reinspection is needed.

After Each Inspection

The tracking stage occurs immediately upon completing each inspection (i.e., exiting inspection step 7 in Figure 2). Inspection results are consolidated, typically by inspection type (e.g., requirements, design, code) to provide management up-to-date insight into project net-savings and ROI.

Other inspection metrics like defect density and defect fix counts can provide an early warning of defect-prone areas and development process weak spots. Consolidating post-inspection data also provides insight into the level of inspection participation by each development area (e.g., requirements) across the project, which can be compared with the planned number of inspections computed during the project inspection planning stage.

Throughout Development and Testing

Finally, there needs to be a means to calibrate whether a project's inspection objective to find and fix defects close to their point of insertion is being achieved. Or, to present this idea as a question: Is the project's inspection implementation improving its early defect removal ability from release-to-release, or has defect removal degraded and (in turn) bogged down testing?

To address this question, quality goals for early defect removal should be established (during planning) that are similar to the typical profile of defect insertion, sometimes referred to as "defect potentials" [3]. A TRW study found that 80 percent of product defects are inserted prior to coding (52 percent during requirements and 28 percent during design) [4]. The project goals for defect removal should strive to remove defects at the same rate as they are being inserted into a product (inspections were designed for this). Defect removal goals should be established for each development phase and recorded in the project's quality plan.

To measure the effectiveness of inspections, actual defect removal data collected from each development phase should be compared to the quality plan goals for early defect removal. To accomplish this, a defect removal measurement tool should be employed that is designed to collect, on a phase-by-phase basis, the:

- Current or past defect removal profile.
- Defect insertion (potentials) profile.
- Project defect removal goals.
- Actual defect removal metrics from inspections, testing, and any other activities.

The defect removal measurement tool could then compare the phase-by-phase defect removal goals versus actual defect removals so management can track release-to-release goal attainment or degradation. The actual defect removal phase profile (from a release) is then used to help set the defect removal goals for the next release.

Inspection Tool Features

Figure 4 provides an overview of the key features previously discussed that management inspection tools should possess.

Summary

Inspections can live up to their potential and be embraced by the development community if:

- Inspections are integral to a well-defined SDLC infrastructure, supported by upper management in each phase of development.
- Computerized management tools are available to assist in planning project inspections and estimating net project savings before commitment to inspections.
- Computerized management tools are available for monitoring inspection process conformance, tracking resulting benefits for multiple inspections, and measuring (calibrating) the actual defect removal effectiveness of inspections.
- Project management and inspection practitioners are provided with training tailored to their unique inspection responsibilities.
- Computerized practitioner tools are available to guide inspection teams for consistent, correct, and repeatable inspection execution (as shown in Figure 5), and to be the basis for management monitoring, tracking, and measurement phase responsibilities.◆

References

1. Priven, Lew, and Roger Stewart. "How to Avoid Software Inspection Failure and Achieve Ongoing Benefits." CrossTalk Jan. 2008.
2. Software Engineering Standards Committee of the IEEE Computer Society. "IEEE Standard for Software Reviews, Section 6. Inspections." IEEE Std. 1028-1997, 1997.
3. Jones, Capers. "Measuring Defect Potentials and Defect Removal Efficiency." CrossTalk June 2008.
4. McGraw, Gary. "Making Essential Software Work." Cigital, Inc. Mar. 2003.

About the Authors

Roger Stewart is co-founder and a managing director of the Stewart-Priven Group. He is an experienced lead systems engineer and program manager in both government and commercial systems, including systems engineering, software development, system integration, system testing, and process improvement. Stewart has a bachelor's degree in mathematics from Cortland University.

The Stewart-Priven Group
7962 Old Georgetown RD STE B
Bethesda, MD 20814
Phone: (865) 458-6685
Fax: (865) 458-9139
E-mail: spgroup@charter.net

Lew Priven is co-founder and a managing director of the Stewart-Priven Group. He is an experienced executive with a management and technical background in system and software development, software quality training, management development training, and human resource management. Priven has a master's degree in management from Rensselaer Polytechnic Institute and a bachelor's degree in electrical engineering from Tufts University.